DIG64: Accelerating the Development of Itanium® Processor-Based Solutions

The Developer's Interface Guide for IA-64 Servers (DIG64) establishes a new industry definition of compatibility for IA-64-based server solutions. This management overview provides an overview of the DIG64 and shows how adopting it can help developers manage legacy technologies and get to market fast with robust, interoperable solutions based on the new Intel® Itanium™ processor.

A Baseline for a New Generation of Server Solutions

Intel's IA-64 architecture and the Intel® Itanium™ processor offer a rare opportunity for the industry to intelligently and collaboratively manage the transition to a new computer architecture. The new architecture creates an opportunity for the industry to work together to:

- Lay the foundation for high-performance, scalable new platforms and a vibrant marketplace of reliable, interoperable server solutions.
- Remove obsolete technologies and address issues that could cause legacy problems as the architecture continues to evolve.

The Developer's Interface Guide for IA-64 Servers (DIG64) addresses these opportunities. Produced by an industry group founded by Compaq, Dell, HP, IBM, Intel, NEC and Siemens and involving dozens of additional companies, the DIG64 outlines baseline system building blocks and software interfaces for servers based on the Itanium processor and subsequent members of the IA-64 family.

Instead of requiring developers to each spend countless months trying to converge on a set of common interfaces from among the hundreds of possible combinations of industry standards and initiatives, the DIG64 specifies a common set of components, interfaces and implementation requirements. Hardware vendors will know what software interfaces, OS and firmware vendors will be using and software vendors will know what interfaces will be in the platform. From this, the DIG64 release 1.0 also lays out a road map for migrating legacy technologies out of the new generation of servers.

Benefits for Developers and Customers Alike

By defining common building blocks and interfaces and proactively addressing legacy issues, the DIG64 provides an array of benefits for both developers and IT organizations. For developers, the DIG64:

- Increases the efficiency of the design process, freeing developers to focus design resources on features that add unique value and differentiate their products in the marketplace.
- Gives firmware and OS vendors a known set of interfaces to build to, enabling them to confidently develop their products concurrently with the hardware and shrink times to market. Hardware vendors benefit from knowing a variety of software applications will run on their platforms, and software vendors benefit by having confidence of a single set of interfaces will work across many systems regardless of vendor.
- Provides a technology migration road map that extends the planning horizon for developers and allow them to create designs with increased longevity.

These benefits extend to business users and information technology (IT) departments:

- Standard interfaces and interoperable layers enhance the reliability and robustness of the resulting servers as well as lowering qualification costs.
- DIG64 common interfaces allow platforms to support multiple operation systems, adding flexibility and investment protection for IT departments.
- Since developers find it easier to build components that work together, customers can enjoy a greater choice of robust, innovative server solutions, with the confidence that interoperability has been tested.
- Because the DIG64 addresses the migration of support-intensive, obsolete technologies to newer, more robust choices, IT departments can control or reduce the costs of server support.

The DIG64 in Brief

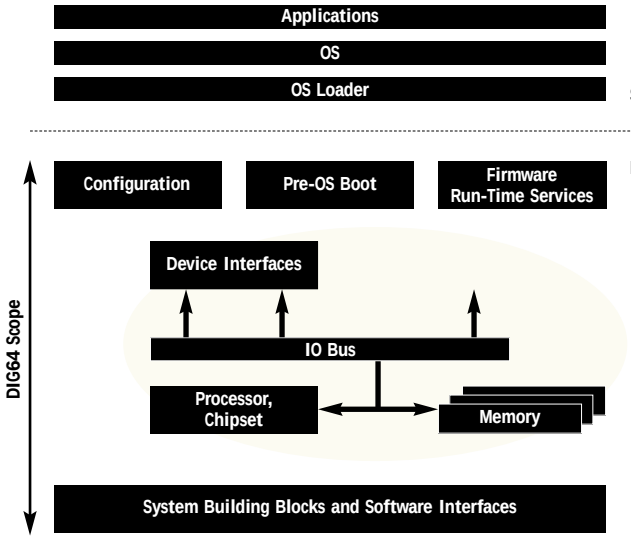
The DIG64 specification defines basic system building blocks, interfaces and programming conventions between IA-64 based servers and system-level software such as the operating system and firmware. The specification covers:

- Core system components such as the processor, chipset, memory, I/O bus and server management hardware.
- Interfaces to peripheral devices for networking, communications and storage.
- Low-level firmware interfaces to the operating system for system configuration, boot and runtime services.

The guide does not create new standards and interfaces but selects components and interfaces from already existing technologies. To ensure interoperability, the DIG64 also specifies implementation requirements for each specification or standard.

The DIG64 spells out a road map for eliminating obsolete technologies and replacing them with newer technologies. Release 1.0, which is available, pertains to servers based on the Itanium processor. Subsequent versions will address future processors as they are developed. A three-level hierarchy of required, recommended and optional guidelines—gives vendors latitude in implementation.

Scope of the DIG64 R1.0



Firmware Interfaces
(Configuration, Pre-OS, OS Run-time, Platform)

Core System components
(Processor, Chipset, Memory, I/O Bus, Server Management)

Component Interfaces
(Networking, Communications, Storage)

Technology Migration
(Legacy Technologies, Replacement Technologies)

The DIG64 is operating-system independent, promoting cross-platform interoperability among servers running Microsoft*, Novell*, Linux*, and UNIX* 64-bit operating systems.

Key Elements of DIG64-Compliant Servers

- Operating system independent
- >4GB memory addressability
- Designed from the ground up for multiprocessing
- Follows a road map for removing obsolete technologies and introducing new technologies
- Firmware creates an abstraction layer between OS and hardware
- Uniform interrupt handling
- MCA error handling
- ACPI-based configuration

The Challenge of Legacy Technologies

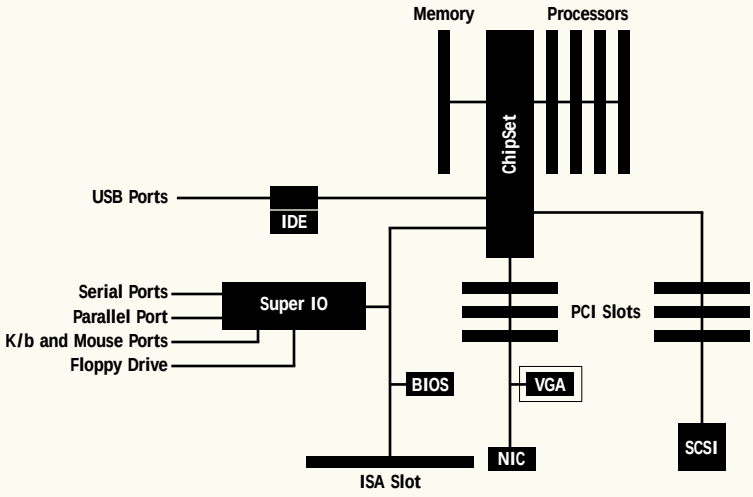
The term legacy refers to out-of-date technologies retained solely to support compatibility with older technologies. These technologies, while useful and innovative in their time, no longer deliver the performance, scalability and robustness that today's enterprise servers demand, and they often reduce the advantages of new technologies. For example:

- Most servers are still configured with standard floppy disks, although modern applications produce files that far exceed their storage capacity.
- Most servers have serial ports that are barely adequate for conventional modems and don't support high-speed adapters at all.
- Most servers include parallel ports to support printers, although printers are rarely connected to servers today.

Many older technologies contain fixed addresses and/or static interfaces, and cause performance bottlenecks and can reduce reliability. For example, legacy I/O buses and devices based on the ISA standard:

- Have slower clock rates and narrow data paths that limit scalability.
- Include timing dependencies that must be managed and validated each time the system processor speeds up.
- Lack the dynamic resource allocation facility needed to support hot-plug capability.

Legacy Design Diagram



- Use non contiguous memory maps and fixed addresses that mire developers in programming details and push validation costs higher.
- Feature oversized slots that consume precious real estate.

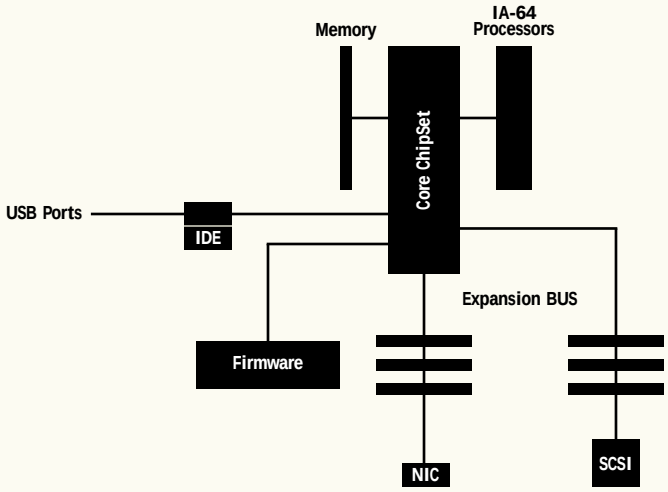
Carrying such technologies forward to the IA-64 architecture would add costs and complexity to a new generation of servers. It also would stifle innovation and make the new servers more fragile by forcing developers to work around the limitations of the old technologies when they add new capabilities. By having industry agreement on migration of legacy technologies, individual developers don't have to be concerned that operating systems or applications will still be trying to access these older technologies.

Phasing Out Obsolete Technologies

Recognizing that *legacy* technologies can't be mandated away in one fell swoop, the DIG64 provides a road map for phasing out legacy technologies over time. The DIG64 also leaves developers free to decide the rate at which they discontinue their support for legacy technologies. It does this by identifying three levels of features for compliant systems:

- Required features must be implemented to comply with the DIG64.
- Recommended features need not be implemented, but implementation is encouraged.
- Optional features need not be implemented, and if they are implemented, must not impact required features. There should be no dependencies on optional features by other architectural levels.

DIG64 Design Diagram



DIG64 Legacy Removal Road Map

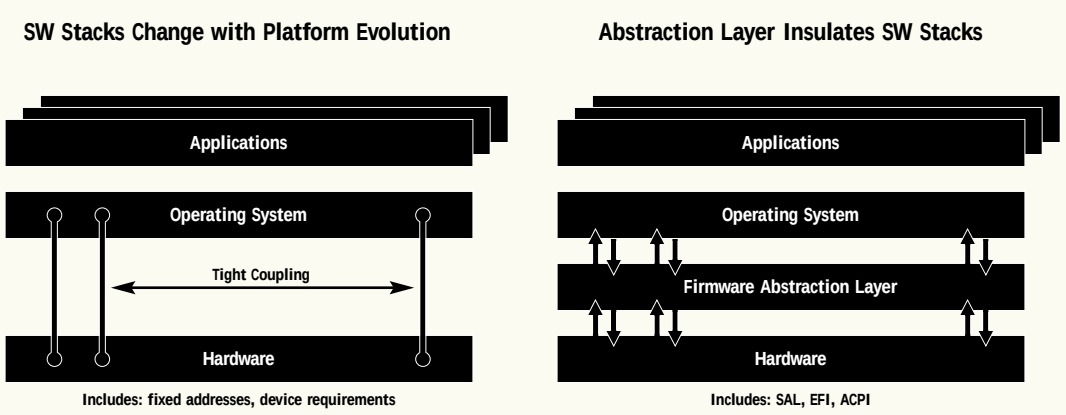
Legacy Technology Guidelines	DIG64 R 1.0			Next release of the DIG64[1]			Replacement Technology
	Req	Rec	Opt	Req	Rec	Opt	
ISA Expansion Slots not Included or Supported	•			•			PCI
System Must Not Include Embedded ISA/LPC Network Adapters, Storage Controllers or Graphics Adapters	•			•			PCI
Support EFI Boot Loader for 64-bit OS	•			•			N/A
Option ROM Support Using EFI		•		•			EFI
Using EFI as the Pre-boot Environment		•		•			N/A
Architectural Support for Non-EFI Option ROMs			•			•	
Architectural Support for DOS			•			•	
Architectural Support for IA-32 Operating Systems			•			•	64-bit Operating Systems
Architectural Support for Windows 95 and Windows 98			•			•	None
Architectural Support for Serial COM Ports			•			•	USB
Architectural Support for Parallel Ports			•			•	USB
Architectural Support for PS/2 Ports			•			•	USB
Architectural Support for Legacy Keyboard Emulation			•			•	
Architectural Support for Floppy Disk Controller on SIO			•			•	
Architectural Support for 8042 Keyboard Controller			•			•	USB
Architectural Support for 8259A PIC			•			•	SAPIC
Architectural Support for 8254 Timer			•			•	
Architectural Support for Real-Time Clock Direct Accesses			•			•	EFI
Architectural Support for CMOS			•			•	EFI
Architectural Support for VGA			•			•	EFI
Architectural Support for Legacy Fixed-Address Functions			•			•	

[1]This column is a statement of direction, not a commitment to the content of future DIG64 releases.

Insulating the OS from the Hardware

Another problem with many legacy technologies is that they often involve hard linkages among the hardware, operating system and platform firmware. This produces a delicate balance where modifications to one server component can cause a ripple of changes that make systems less reliable and drive up development and qualification costs. The tight coupling also slows the pace of innovation, because the development of different system elements often can't proceed until other elements are known. And it requires peripherals and platform OEMs to customize their products for each operating system, again adding to development costs.

DIG64: Firmware Abstraction Layer



Along with identifying legacy technologies that are candidates for removal, the DIG64 provides a robust new firmware model that creates an abstraction layer between the hardware and the operating system and further assists in removing obsolete technologies. By abstracting the hardware from the software, the firmware model enables innovation to occur in different components without negatively affecting others.

This layer of abstraction serves to buffer both the OS and the platform from legacy concerns. For example, the operating system doesn't have to know whether the platform's serial I/O capability is a standard PC/AT serial port, a serial port attached to a USB interface or some other character-based serial I/O device; it can simply write to a "Serial I/O Port" abstraction. The firmware model includes a boot loader and preboot environment that allow the system hardware and operating system to communicate at boot time and initialize the system with current information.

The firmware model is OS-agnostic, giving hardware developers a uniform OS interface whether the server is to run UNIX*, Microsoft* or other OSs. The firmware model enables a single platform to boot multiple 64-bit operating systems, increasing the customer's freedom of choice and reducing the burden on developers who want to support multiple operating systems.

Key elements of the firmware model include:

- Advanced Configuration and Power Interface (ACPI)—Allows robust, simplified system configuration.
- Extensible Firmware Interface (EFI)—Manages boot and runtime services, provides an abstracted interface to the OS and enables technology migration.
- Machine Check Abort (MCA)—Provides robust error checking.
- OS loader—Handles pre-boot initialization between the operating system and the hardware.
- System Firmware Layer—Abstracts the platform's implementation-specific features from the operating system.

Act Now!

No one wants to reinvent the wheel—or get left behind when the industry moves forward. The DIG64 V1.0 sets the industry standard for Intel Itanium processor-based server solutions. Many vendors already have announced that their next-generation designs will comply with the DIG64. If your company isn't one of them, now is the time to act. Go to **http://www.dig64.org** to get details about the DIG64 guidelines. Then, become a DIG64 adopter. You'll have the opportunity to participate in upcoming DIG64 interoperability events and promote your DIG64-compliant products at DIG64 events and on the DIG64 web site. Best of all, you'll be able to take advantage of the benefits the DIG64 provides—the ability to make your design process more cost-effective and efficient, shrink time to market and deliver robust, interoperable designs that maximize innovation and minimize legacy headaches.

Don't delay. Adopt the DIG64 today, so you—and your Itanium processor-based server customers—can benefit.

www.dig64.org